

DOCUMENT RESUME

ED 116 954

SE 020 189

AUTHOR Pleck, Michael H.
 TITLE On the Role of Computer Graphics in Engineering Design Graphics Courses.
 PUB DATE Jun 75
 NOTE 34p.; Paper presented at the Annual Meeting of the American Society for Engineering Education (Colorado State University, Ft. Collins, Colorado, June 16-19, 1975)
 EDRS PRICE MF-\$0.76 HC-\$1.95 Plus Postage
 DESCRIPTORS College Science; *Computer Assisted Instruction; Design; *Engineering Education; *Graphic Arts; Higher Education; Instructional Materials; *Undergraduate Study

ABSTRACT

The implementation of two- and three-dimensional computer graphics in a freshmen engineering design course at the university level is described. An assessment of the capabilities and limitations of computer graphics is made, along with a presentation of the fundamental role which computer graphics plays in engineering design instruction. (Author/CP)

 * Documents acquired by ERIC include many informal unpublished *
 * materials not available from other sources. ERIC makes every effort *
 * to obtain the best copy available. Nevertheless, items of marginal *
 * reproducibility are often encountered and this affects the quality *
 * of the microfiche and hardcopy reproductions ERIC makes available *
 * via the ERIC Document Reproduction Service (EDRS). EDRS is not *
 * responsible for the quality of the original document. Reproductions *
 * supplied by EDRS are the best that can be made from the original. *

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY.

Event Number 3250

AMERICAN SOCIETY FOR ENGINEERING EDUCATION

ANNUAL CONFERENCE, JUNE 16-19, 1975

COLORADO STATE UNIVERSITY

FT. COLLINS, CO. 80521

ON THE ROLE OF COMPUTER GRAPHICS IN ENGINEERING DESIGN GRAPHICS COURSES

Michael H. Pleck, Ph.D.
Assistant Professor in General Engineering
Research Assistant Professor in the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Abstract

The implementation of two and three dimensional computer graphics in a freshman engineering design graphics course at the University of Illinois is described. Within the context of this experience, an assessment of what can and cannot be achieved with computer graphics is made. As a result, the fundamental role of computer graphics in engineering design graphics courses is examined.

ED116954

SE 020 189

INTRODUCTION

Beginning in the late 1960's computer graphics became popular and attractive in undergraduate engineering education and specifically in engineering undergraduate design graphics courses. However, the role played by academia was one of a follower rather than a leader for it was largely the growing industrial, governmental and research use of computer graphics that attracted the attention of educators. Many factors entered into this educational lag. Perhaps the most notable were: the unavailability of computer graphics equipment in schools for general education due to a lack of resources; the lack of a transfer of knowledge from specialized research and application settings to the classroom; the rapid and often confusing proliferation of display devices and technologies that often required an expert's knowledge to make use of them; and the lack of coordination between the computer graphics hardware and software and the needs of the users which in some cases were not yet well defined because of the rapid technology-led impact.

With the situation noted, no major computer graphics textbook appeared until 1973 [1], albeit intended for advanced undergraduate and graduate study. The literature abounds with state of the art advancements, but little has appeared for undergraduate student audiences. Some useful articles and book chapters have managed to appear (see, for example references 2 - 7), but they often left questions unanswered and seldom presented computer graphics programs that could be put to use.

In addition to obtaining knowledge through the literature, the computer graphics educator could attend workshops and short courses. The most successful of these has been a continuing University of Michigan summer conference [8]. It has treated computer graphics from both a computer scientist's and designer's point of view. However, the general level of presentations was aimed at the specialist who would transfer his learning into research and applications programs. Other lesser known short courses treated specific topics of engineering design graphicists but were often publicized little or only one-time offerings.

During the same period, computer graphics firms produced commercial program packages for basic and applied graphics (for example see references 9 - 12). But these tended to be designed around needs that they, and not the eventual users including educators, envisioned.

Recognizing the state of affairs, the Computer Graphics Committee of the Engineering Design Graphics Division was formed. Notably, in 1972 it was responsible for organizing a computer graphics summer school intended specifically for engineering educators. The school attempted to place hardware, software and usage into an acceptable framework for education, and engineering design graphics courses in particular. At this workshop, basic two-dimensional plotting via Fortran [13] and the TRIDM [14, 15]* program for the projection of three-dimensional objects were presented in detail and put in the public domain.

This bootstrap effort was aided by much unheralded and unpublicized effort and cooperation. As a result, the engineering design graphics profession is now cognizant of the potential of computer graphics or already using it. It would be an arduous task here to survey the current state of its use in engineering design graphics courses. Perhaps that would be a suitable project for an Engineering Design Graphics Division forum. Let it suffice to say that it is this author's knowledge that many institutions are now using computer graphics in engineering design graphics and related courses that were not doing so five years ago.

In the preceding paragraphs, an historical perspective of the use of computer graphics in undergraduate engineering design graphics courses has been given. With this perspective the development and implementation of computer graphics at the author's institution since 1968 will be described and evaluated. Finally, with a realization of some current engineering trends affecting engineering design graphics, the role of computer graphics in engineering design graphics courses will be examined.

*TRIDM is a variation of DRAWL-70 [16]

LOCAL PROGRAMS DEVELOPED

Aware of the educational state of the art in 1968, the Department of General Engineering at the University of Illinois, Urbana-Champaign, undertook a modest effort to incorporate computer graphics in the engineering design graphics classroom. The initial philosophy was to develop two computer graphics programs for student instruction use. The University computer center was to be used for implementation of the programs in batch mode. The host computer was an IBM 360/75 and the plotter device, run online, was a CalComp 765/drum plotter using eleven inch paper. Anticipating large scale use and without the availability of interactive graphics displays, the batch mode with card input was deemed to be an acceptable student use format -- by default!

Briefly, the two programs envisioned were as follows. The first was to be drafting oriented with plotting commands that permitted various geometric elements to be sequentially combined on the plotter surface. In it the student was to use the commands much like one used conventional straightedges, templates, and a pencil to create a drawing. The second was to deal with three dimensional constructs and their subsequent projection onto a viewplane. This program would require the student to think in three dimensional space, assembling an object out of predefined or user specified primitives that could be scaled, rotated and translated into position. Subsequently, the student would request one or more images of the buildup object in which the orientation of the viewplane and the object could be specified. The objects would appear as "wireframes" without attempts to solve visibility.

Both programs were organized in subroutine fashion and written in Fortran IV. Roughly speaking each subroutine or its entry points corresponded to a user command. After a crash effort, initial versions were put to the classroom test in 1968. As a result of this testing improvements in the capabilities and function of the programs were made from 1968-70. The most important of these centered around the avoidance and suppression of abnormal program terminations (abends) that are so ubiquitous in IBM-based

CalComp plotter software. Another was the development of an interpretive monitor that allowed drafting programs to be batched through the use of interpretive command statements that replaced the Fortran ones.

Evolutionary sidelights put aside, the programs as currently used will now be briefly described, together with sample input-outputs.

GRAFIX Program

The GRAFIX computer program [17] is an instructional drafting-aid program utilizing computer graphics on a digital computer/plotter system. It consists of three essential parts: drafting routines, drafting monitor, and an installation-oriented plotter package.

The Fortran based drafting routines permit the placement of lines, circles, ellipses, annotations, etc. on the plotter paper. They may be thought of as counterparts to a draftsman's templates and instruments. They are brought into play by selecting the codeword for the appropriate geometric element and by specifying associated arguments. Together, the codewords and arguments comprise "commands" which are in an easy-to-use format, much like Draft-Aid [10], PAL [11] and XYMASK [18].

Specifically, the input commands are in the form CODE ARG1, ARG2,..... where CODE denotes a coded drafting routine name and the arguments ARG1, ARG2,...represent a sequence of numerical values or an annotation. The arguments are needed by the drafting routine to produce its output onto the plotter. Typically these are coordinates, angles, distances, notes and the like or flags to specify the desired options. As an example the command

P 2.0, 2.5, -2, *THIS IS A STATEMENT

specifies a simple plotter pen movement. The coordinates to which the pen moves are the first two arguments, while the third is a pen control ("down on paper") and reference point flag ("make a new reference point"). The reader will probably recognize it as the Fortran based CalComp plotter command CALL PLOT. About another dozen and a half commands comprise the repertoire of GRAFIX. They are detailed in Appendix 1.

The drafting monitor, the second component of the GRAFIX program, exercises overall control over the computerized drafting process. It is written in Fortran IV and IBM 360 assembly language. It establishes paper boundaries from user-supplied information, interprets the drafting commands, checks the commands for format and value errors, applying fixes where possible, and directs program control to the appropriate drafting routine. Significantly, the monitor employs a windowing concept which permits crossing of the plotting boundaries without program termination. The drawing simply continues when the boundaries are recrossed. Moreover, the monitor permits the batching or concatenation of many user programs each separated by a user identification/control card. This feature of the monitor permits GRAFIX to make efficient use of the computer facilities.

The third component, the plotter package, consists of software that maintains control of the plotter paper coordinate system and records of plotter paper and time used. In addition, it contains software that is associated with the conversion of desired drafting lines into the lines on the plotter paper. The plotter package currently used by this version of GRAFIX is for digital incremental pen plotter, so the drawings are physically provided by a moving pen whose motions can be thought of as analogous to a draftsman's. It is a locally modified and extended set of CalComp basic plotting routines.

Two examples of GRAFIX usage are shown, a multiview orthographic drawing and an isometric drawing. Figure 1 shows the input card deck facsimile and the corresponding program listing for the multiview drawing in Figure 2. There are numbers adjacent to various lines in Figure 2. They were added by hand to permit the reader to associate the lines drawn to the command numbers found in the left hand column of the printed output in Figure 1. Figure 3 shows the program listing and plotter output for an isometric drawing of the same bracket (Figure 2). Again the handwritten numbers correspond to the command numbers. An oblique drawing could have been produced with about the same effort. Some subjective observations about the usefulness and student reactions to this program will be found in a subsequent section.

01 SEP 71

CRAFIX EXAMPLE #1

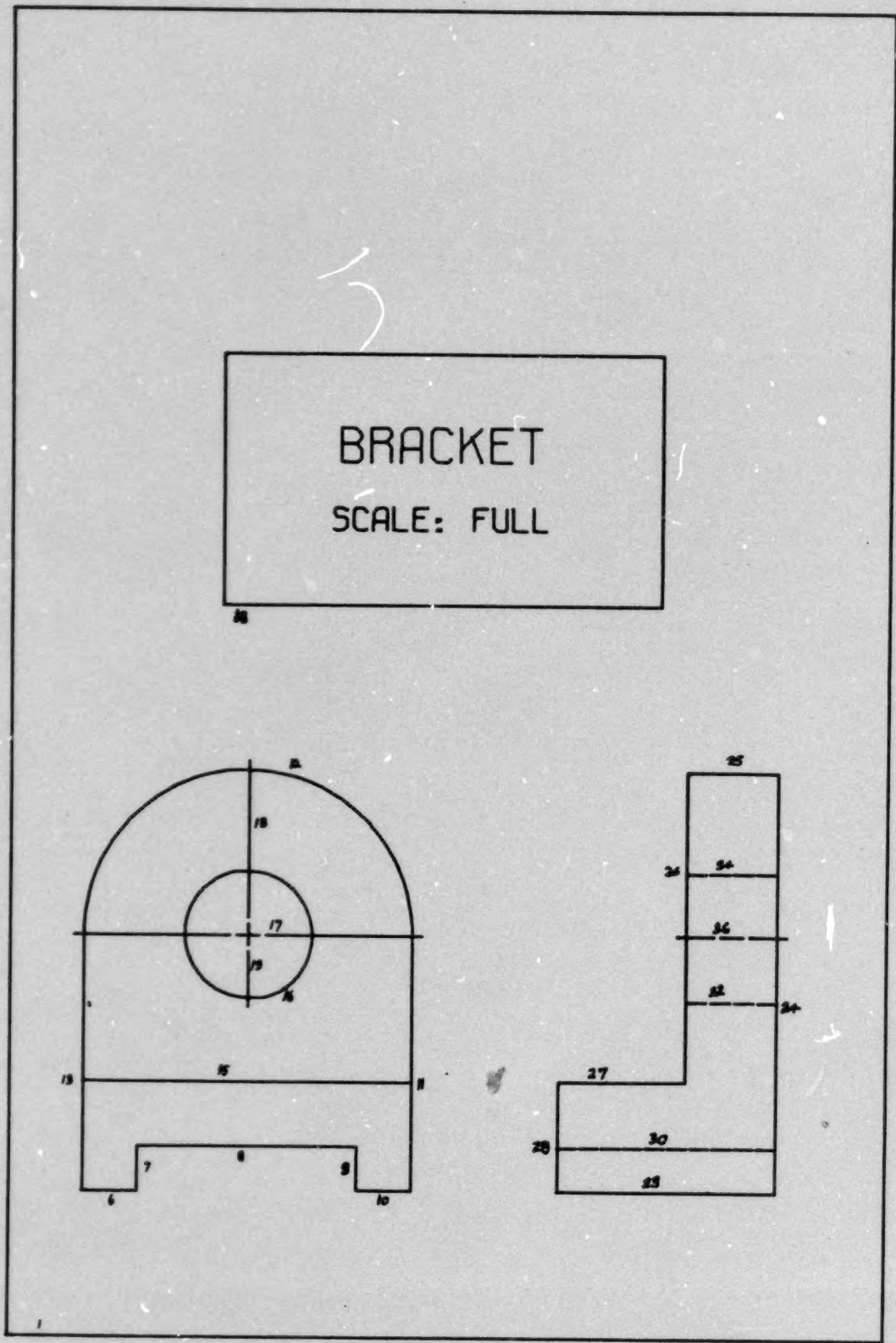


FIGURE 2



GATRAN Program

GATRAN [19] is a package of Fortran subroutines that permit the imagined construction of three dimensional objects. Further, it allows their movement behind a fixed observation plane upon which views may be projected. It is like other "stick figure" three dimensional display systems [2 - 5, 14], for they all are based on coordinate transformations. But GATRAN does have some interesting educational features. Its use is broken down into roughly three stages: definition, construction and viewing.

In GATRAN objects may be defined in two ways. The most general method involves the specification of the points and lines associated with an object. The second method of defining an object is implicit. That is, the necessary point line and line combinations are contained in the memory of the computer, having been put there in advance. Both use coordinates local to the object defined. Each predefined object is reached via a routine that bears its name. Two such routines are presently contained in GATRAN. One is for right rectangular prisms, the other for approximations of right circular cylinders and cones. Each requires certain parameters to define the dimensions of the body and its location. With these, they are capable of calculating the coordinates of the points. Each defined object can then be put to use in the construction of a composite object. The construction is not physical, but rather the imagined placement of elements in a three-dimensional cartesian coordinate system. In reality, points take on new coordinates that are the result of rotations, translations and scalings of the defined parts within a global or overall coordinate system. These are computed and stored for later reference. The lines of all the elements comprising an object to be viewed are stored in succession for later recall by drawing routines. It is to be noted that only one or many bodies may be included in a constructed object and that a given body may be added to it many times.

GATRAN provides two types of views of the object which has been constructed. The first is orthographic. It includes orthographic views in the conventional sense, as well as axonmetric projections. The latter

views are obtained by turning the object so that its principal faces are inclined to the viewing plane. In both, the projection lines are parallel to each other and perpendicular to the fixed viewing plane. The second type of view available in GATRAN is perspective. By the method of [2] projection lines converge to a sight point and produce the familiar perspective distortions.

GATRAN provides for the use of drawing scale factors. These are not to be confused with the scaling of bodies performed by construction routines. Overall scale factors may be specified directly or indirectly to keep the view within the plotter boundaries. However, the viewing routines do not permit partial or "windowed" views, nor do they allow the ability for an observer to place himself inside the object in perspective views. The simple data structure employed by the program is the so-called disjoint-line method [20] as opposed to strings [20] employed by TRIDM. In the disjoint line method, an ordered set of points in x, y, z coordinates is created. These are subsequently referenced as the end-points of various line segments comprising the stick figure.

Plotting calls are embedded within the GATRAN routines. As in GRAFIX they access the locally modified CalComp basic plotting software.

Complete descriptions of the GATRAN commands will be found in Appendix 2. An example should suffice to show their use in creating an object and viewing it. Figure 4 contains the Fortran statements and data cards to produce the plotter drawings found in Figures 5 and 6. Figure 4 contains the program output listing that, together with the data card listing, illustrates the disjoint-line data structure.

Availability

At present, both GRAFIX and GATRAN are not very portable to other institutions.* Nor do they permit easy modifications such as feature additions. In forthcoming revisions, attention will be directed to using structured programming techniques, providing flexibility for user changes, and establishing as much independency from local software as possible.

*Readers interested in obtaining Fortran-callable drafting routines corresponding to the GRAFIX commands in Appendix 1 and GATRAN should direct their inquiries to the author. Versions based on the standard set of CalComp basic plotting routines will be available at a slight change.

```

C  SAMPLE PROGRAM
C  ..... (CONTROL CARDS NOT SHOWN; OBTAIN FROM INSTRUCTOR)
C  CALL START
C  CONSTRUCT LOWER MODULE SECTION, THEN UPPER MODULE SECTION
C  CALL CYLIN (1.C,2.0,6.C, C.,0.,0., 0.,0.,0.)
C  CALL CYLIN (1.C,1.0,2.4, C.,C.,C., C.,6.,0.)
C  DEFINE A WINDOW PORT WITH POINT 1 ON THE Z AXIS AND
C  A SLANT EQUAL TO THAT OF THE LOWER SECTION (USES DATA CARDS)
C  CALL DEFINE (IPORT,ILPORT,3,3)
C  CONSTRUCT WINDOW ARCLAC LOWER SECTION 4 UNITS ABOVE BASE
C  CALL IMAGE (IPORT,ILPORT,3,3, 1.,1.,1., 45.,0.,0., 0.,4.,0.)
C  CALL IMAGE (IPORT,ILPORT,3,3, 1.,1.,1., 135.,0.,C., 0.,4.,0.)
C  CALL IMAGE (IPORT,ILPORT,3,3, 1.,1.,1., 225.,0.,C., 0.,4.,0.)
C  CALL IMAGE (IPORT,ILPORT,3,3, 1.,1.,1., 315.,0.,C., 0.,4.,0.)
C  PREPARE FLCTER FOR FIRST VIEW (CENTERED VERTICALLY)
C  CALL FLCT (7., 5., -3)
C  DRAW ORTHOGRAPHIC VIEWS WITH OBJECT IN CONSTRUCTED POSITION
C  CALL CFTHC (0.,0.,0., 0.4)
C  ADD AN ENCLOSURE TO ENHANCE A PERSPECTIVE EFFECT
C  CALL BLOCK (6.,9.,6., C.,C.,C., -3.,0.,-3.)
C  OBTAIN PERSPECTIVE VIEW
C  CALL PERSP (15.,15.,15., 2., 4.5,4.5)
C  COMPARE TO (ROTATED) ORTHOGRAPHIC VIEW, SHIFTING ORIGIN FIRST
C  CALL FLCT (C.,-1.7,-3)
C  CALL SINW (15.,15.,15., C.4)
C  OBTAIN TALLY, MAKE RIGHT HAND MARGIN AND END PROGRAM
C  CALL CLINT
C  CALL FLCT (4.,C.,-3)
C  STOP
C  END

```

0.	0.	1.65	POINT 1
.5	.866	1.32	POINT 2
-.5	.866	1.32	POINT 3
1	2		LINE 1
2	3		LINE 2
3	1		LINE 3

Data Cards

Output

```

*****
*      G A T R A N (VERSION 1.2)      *
*  UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN  *
*  DEPARTMENT OF GENERAL ENGINEERING  *
*****

      DEFINED POINTS FOR OBJECT 1
      PT #      X      Y      Z
      1      0.C      C.C      C.165E 01
      2      C.500E CC      C.866E 00      0.132E 01
      3      -C.500E CC      C.866E 00      0.132E 01

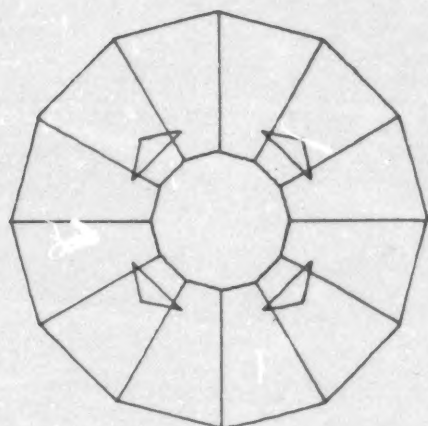
      DEFINED LINES FOR OBJECT 1
      LN #      FROM PT #      TO PT #
      1      1      2
      2      2      3
      3      3      1

      TALLY OF POINTS AND LINES USED SO FAR:
      "DEFINE" POINTS: 3      "DEFINE" LINES: 3
      TOTAL POINTS: 6E      TOTAL LINES: 96

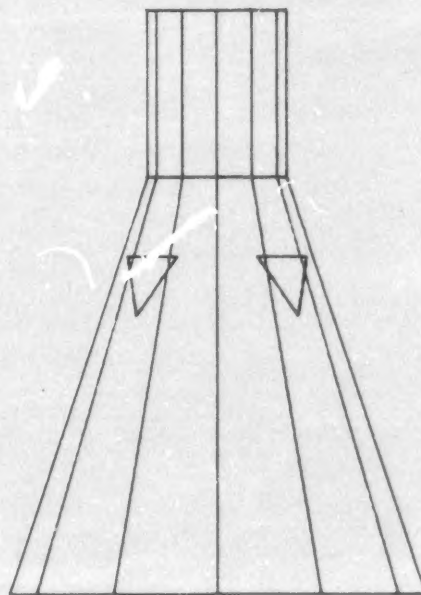
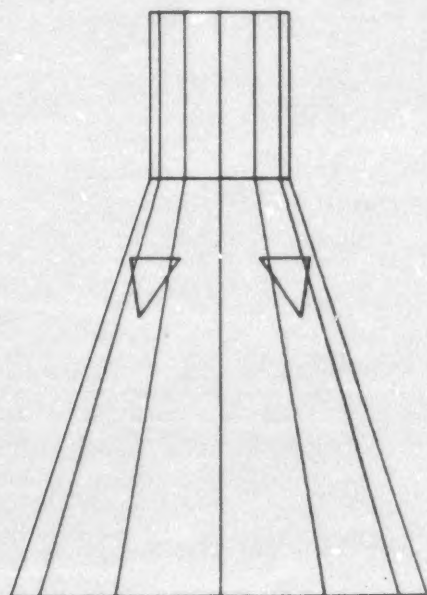
```

FIGURE 4

RY (DEG)	= 0.0
RZ (DEG)	= 0.0
RX (DEG)	= 0.0
SCALE	= 0.400
	(INCH/VALUE)



ORTHO



SINW

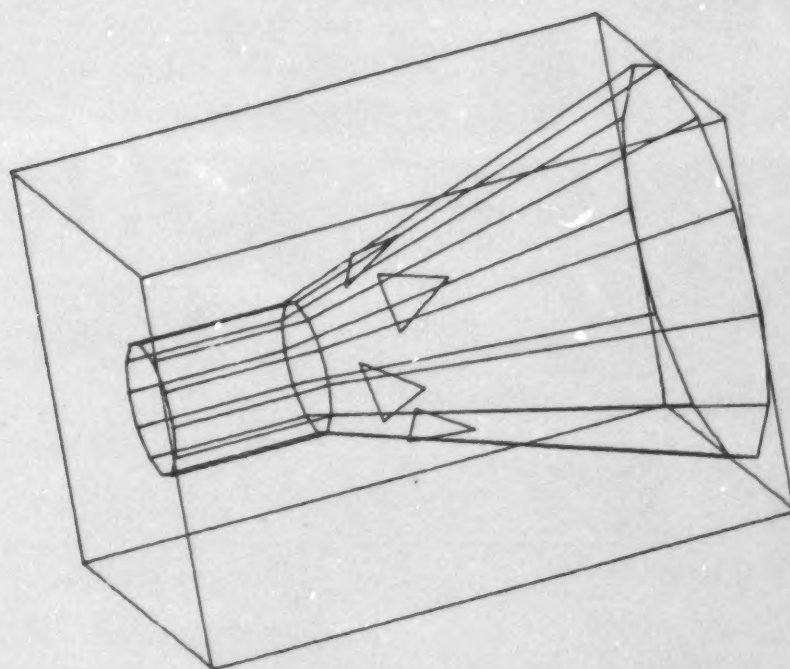
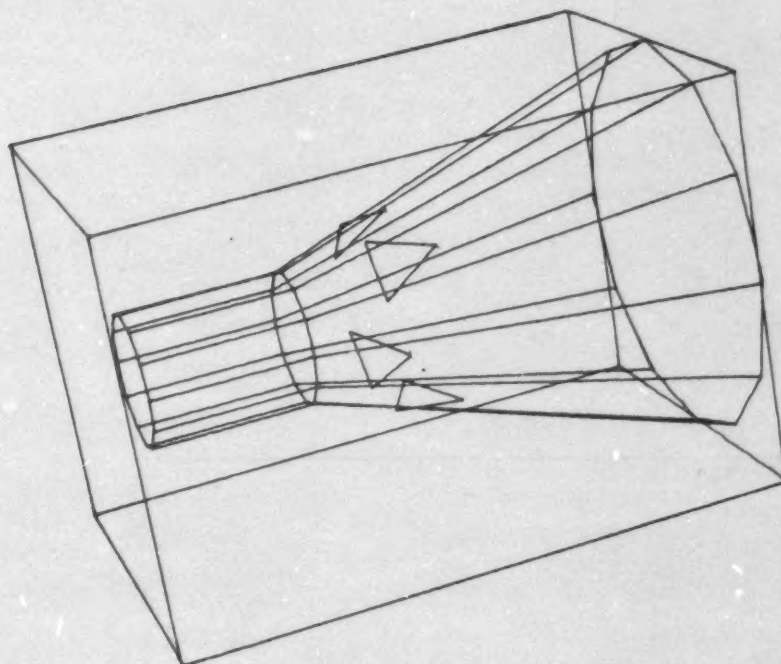


FIGURE 6

PERSP



USAGE AND EVALUATION

The GRAFIX and GATRAN programs have been used in numerous sections of the basic engineering design graphics class, G.E. 103 at the University of Illinois at Urbana-Champaign. At present, the course is required for all engineering freshman except in electrical engineering. Students are not expected to have had prior computer experience. All students taking G.E. 103 are required to complete two GRAFIX projects, one an orthographic multiview drawing, and the other a pictorial -- isometric or oblique. On the other hand, GATRAN has been used by less than half of the students because of the drain on resources and computer funds created by the submission of a large number of individual plotting programs. It will be recalled that GRAFIX has a batching feature which tends to minimize these costs.

GRAFIX Program

The experience with the GRAFIX program may be characterized as modestly successful. Students seem to really enjoy having the opportunity to generate computer assisted drawings. There is a certain amount of interest that naturally seems to stem from a mixture of students, computers and drawings. But beyond the mere involvement, the students are having to put their understanding of basic orthographic and pictorial engineering drawings to the test. For the computer assisted drawings can only be as correct as the student's understanding permits. There is no magic that covers up deficiencies. The net effect seems to be an increased self study to avoid making technical errors.

But at this point an interesting question may be raised. If one must in effect master traditional theory, what advantage does the computer assisted procedure offer? Certainly considering the time to analyze drawing in terms of GRAFIX commands and to punch computer cards, there is probably no advantage at all! In fact there is a point of no return where using GRAFIX can become sheer drudgery! However, GRAFIX was designed to illustrate the types of industry programs available. As such it makes no pretenses to contain some of the niceties of industry programs that would implicitly require the student to have programming skills. Granted, at this point in

time, it might have been designed differently, but that is not the issue. In summary, the students are introduced to algorithms, albeit drawing oriented. They successfully master a basic set of drafting commands akin to industry practice. Finally they bolster their understanding of engineering drawings.

GATRAN Program

The GRAFIX program produces lines in much the same sequence a draftsman would. On the other hand, GATRAN provides an experience in three dimensional space relationships and visualization with views produced more or less automatically, although without proper visibility. This was its intention. The ability to think in three dimensional terms is taken for granted by many, once mastered, but it is not taken lightly by students who are not accustomed to it. What GATRAN and similar programs force the student to do is build a model of some object much in the same way one would do it with actual physical wireframe blocks or other primitives. Point and line relationships come into play in defining primary objects to be used in the construction process. Primary object interrelationships then become important while actually constructing or assembling the object. Moreover, understanding and applying the transformation steps is akin to assembling the components in the real world via automation techniques.

This is not to say that GATRAN is without drawbacks. It and similar programs such as TRIDM must represent actual smooth surfaces as a group of discrete flat surfaces with extraneous edges. The assembled object must be considered as a stick figure or solid requiring visibility determinations. Intersections of primary bodies are not determined.

Assignments with GATRAN have been usually open ended -- and with some startling results. Instead of mere brackets, blocks and other paraphernalia found in graphics texts and workbooks, students have focused upon the memorabilia of ancient Greece to the regalia of 2001: A Space Odyssey and everything in between. "Hah! Engineering graphics?" a critic once scowled. No, not in the traditional sense to be sure. But such drawings

can and do serve as a basis for a further classroom exercise. It hinges around the student producing a so-called technical rendering of the object he has created. In doing so one essentially traces the computer produced drawing. But bumpy line-infested computer models of otherwise smooth surfaced objects are smoothed out and visibility is interpreted by removing lines and surfaces that would be hidden. Finally shading and shadowing and an occasional artistic background are added. Figures 7 and 8 show one such student-conceived exercise.

One would only have to visit architectural and engineering firms to realize that computer graphics is being used identically for presentations and reports. So it is felt that through GATRAN our students get a good dose of self taught three dimensional visualization in a computer assisted technical drawing environment.

WHAT ROLE COMPUTER GRAPHICS?

The author's implementation of computer graphics described in the previous sections is by no means implied to be the best or necessarily even a good approach. It is felt that certain limited objectives were met, including creation of student interest, exposure to concepts of industrial practice, and the development or reinforcement of graphic principles. All these objectives were planned to fit within the existing overall course structure of G.E. 103.

As a result of the author's experiences and those of other educators, the question must now be asked, "What role should computer graphics play in engineering design graphics courses?" But before attempting to answer it, let's take a look at some important current trends and developments within the design profession affecting the use of graphics.

Some Recent Trends

First, consider the practice of engineering design graphics as we know it in the context of the design process. To describe a part for shop manufacture, drawings are the medium of communication. The process whereby this communication takes place, however, is both inefficient and awkward. First, an object, real or imaginary, is mapped into two dimen-

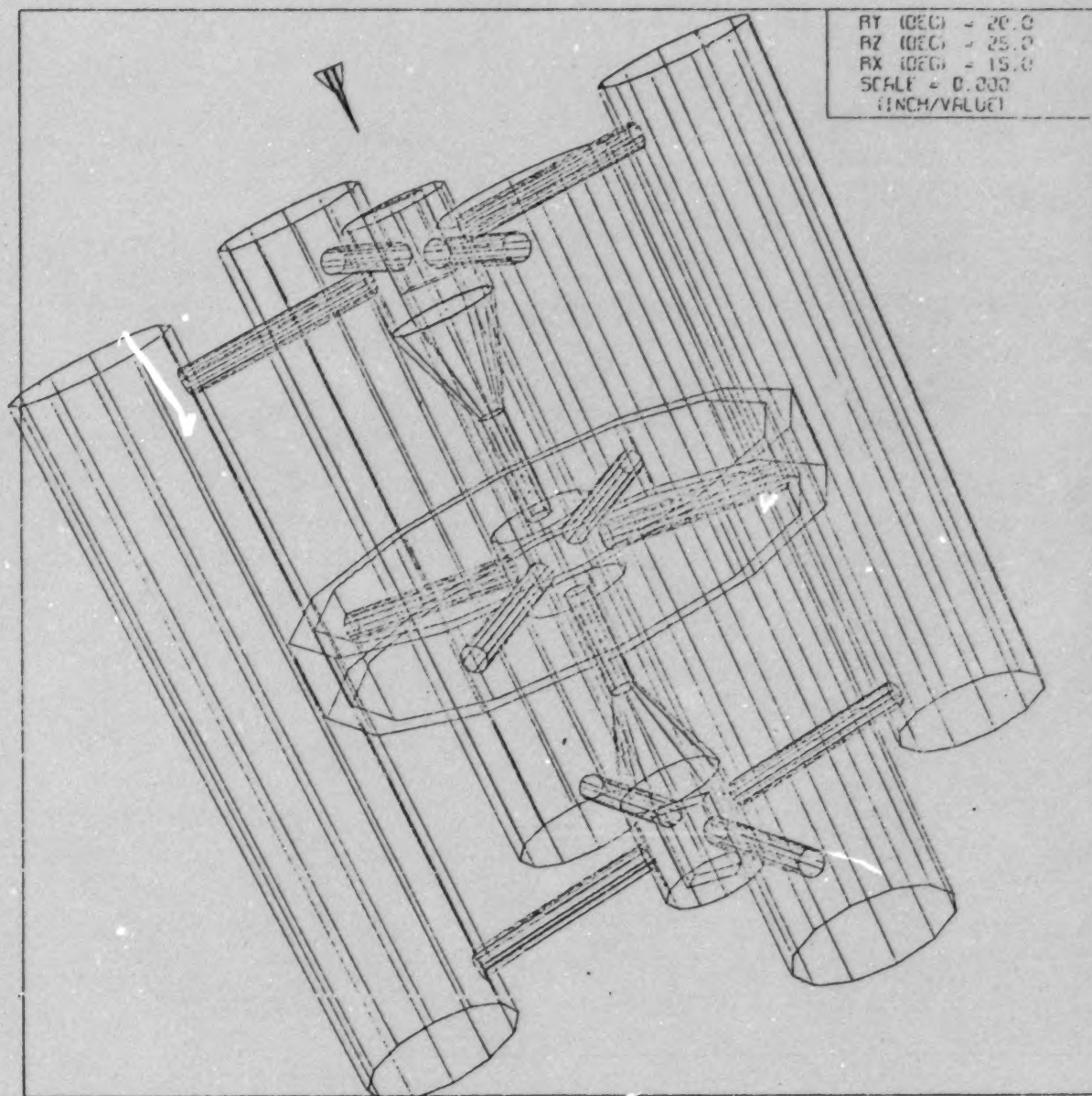


FIGURE 7

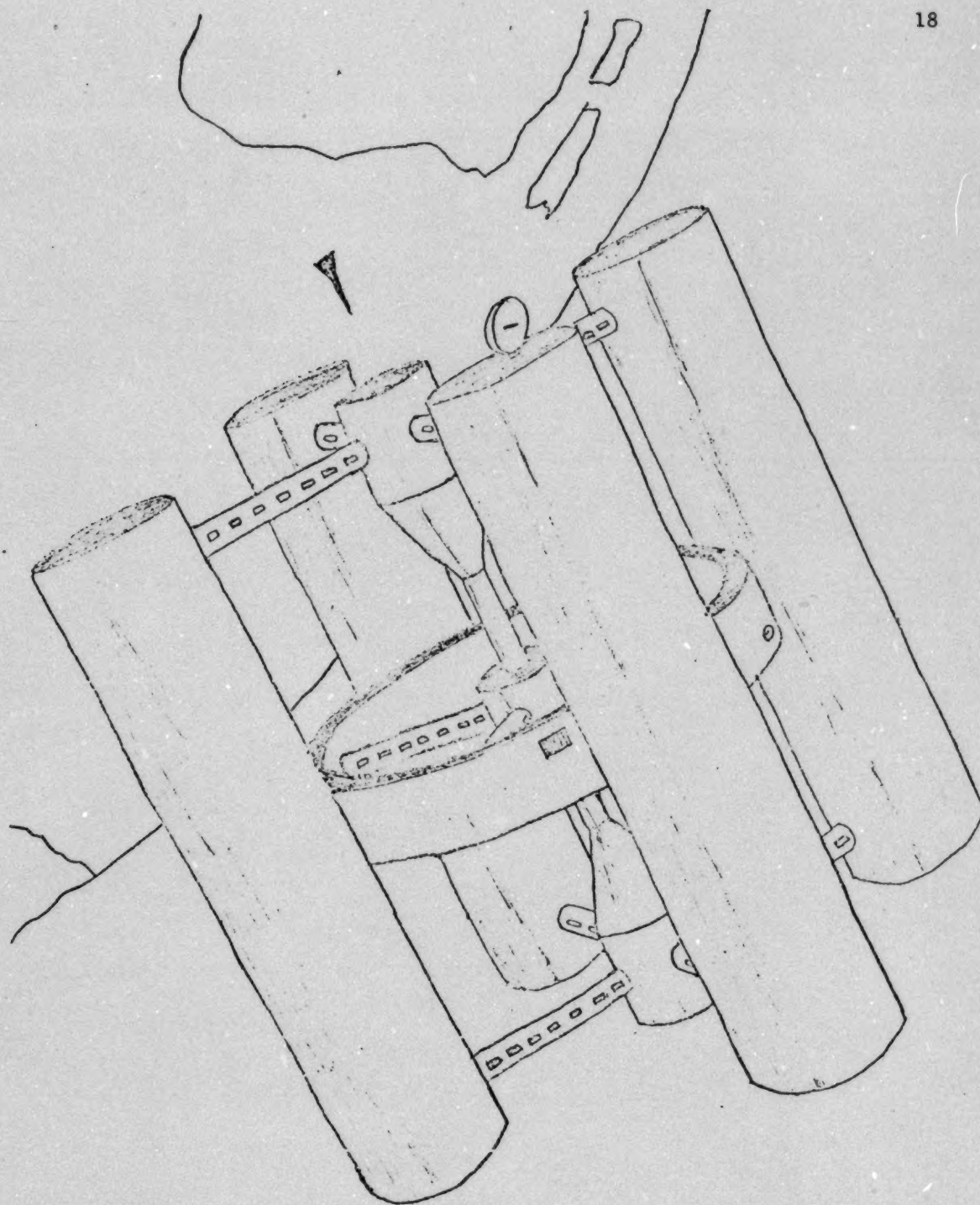


FIGURE 8

sional drawings or layouts, usually orthographic views, by a designer and/or a draftsman. In doing so the designer/draftsman must carefully choose views so that no information is left out or hidden in the projections. Very often auxiliary and sectional views must be employed and careful attention given to dimensioning. Then in order to manufacture the part, a production engineer or machinist must visually reconstruct the object. He must do this so that such things as fabrication processes, machining steps or the like can be determined. Oversimplified? Perhaps. But the role of the engineering drawing should be clear.

Computer graphics, specifically that associated with hardcopy output devices, was seen to be a logical alternative to the production of these engineering drawings and layouts. In some cases, traditional drawing standards had to be set aside because they simply could not be included. But this is not a major issue. The fundamental use of engineering design graphics remained, but with computer assistance.

Certain developments, however, began to attack the traditional ways. Computer aided design (CAD) activities focused upon the use of interactive graphics to allow the visual interaction between a designer and what was being designed [21]. Graphical displays could and did include conventional views. But the computers and displays began to be programmed to produce such effects as dynamically moving images and textured views that appeared very much like camera-recorded objects. Associated with all of this, auxiliary programs could be invoked to analyze the parts for such things as strength, motion and interrelationships. This increased the power of the designer. But at the end of the CAD phase traditional engineering drawings were produced in one form or another for shop fabrication and production of parts.

At the other end of the spectrum, Computer Aided Manufacturing (CAM) activities were becoming firmly entrenched. One of the most prominent of these involved the use of numerically controlled (N/C) machine tools [22] that were programmed with APT [23] or similar commands. Initially the programming was done by a so-called parts programmer who trans-

lated the engineering drawings into machine tool commands punched onto computer cards or paper tape. Over the years, the process has become more sophisticated with the use of magnetic tapes and the direct control of N/C machines by computers [24]. But the input to CAM systems remained the engineering drawing.

Recently, efforts have been undertaken to interface the CAD and CAM activities directly by computer. It should be immediately obvious to the reader what this portends for the conventional engineering drawing in a CAD/CAM environment. Since it serves as the communication link between CAD and CAM, the engineering drawing is no longer necessary for that purpose. Instead, part definitions contained within the computer system are converted directly to manufacturing steps. This is largely developmental for certain highly specialized industries, and has made the greatest inroads in the electronics industry [25, 26] where two-dimensional frameworks prevail. In the more difficult world of three dimensions basic theories of total CAD/CAM systems are being formulated [27].

Goals for Usage

The CAD/CAM scenario depicted does not necessarily spell a quick death knell for engineering drawings. It is unreasonable, despite the emphasis currently being put on automated manufacturing and improved national productivity, to expect that CAD/CAM will sweep out formal engineering drawings industry-wide in a short time. Indeed graphics as a communication medium via sketches, etc. in informal situations shall undoubtedly always remain with us. On the other hand, educational institutions should not ignore the changes integrated CAD/CAM systems in selected industries will bring. It would be regrettable if education were again to assume a role of follower as it did with computer graphics in general. To this end, the title of this paper might have been "On the Present and Future Roles of Engineering Design Graphics Courses." However I have chosen not to prognosticate heavily on the future of graphics. Rather, I have reasoned it more valuable to put the use of computer graphics into a proper frame of reference, given the possible developments in the years to come.

What I propose then, is not a specific set of computer graphics programs. Rather, I propose some goal-oriented concepts. These may be implemented in different ways via existing or undeveloped educational computer graphics programs. The goals, with some broad suggestions for implementation then, are as follows:

1. Permit engineering design graphics course students to experience the conceptual approach to design. That is, instead of translating a mind's eye view of an object, real or imaginary, into two dimensional drawings, permit them to deal in the model space, creating the total design from primitive elements. This can be strictly two dimensional, as in circuit or printed circuit design, or three dimensional, as in machine parts and the like. The student should devote his time to building up the "design" -- dealing with primitive element interrelationships positioning, etc. -- and subsequently interpreting the images produced. Let the computer perform the task of generating the two dimensional images on paper such as GATRAN and TRIDM do.

2. Help the students become aware of the inroads being made by CAD/CAM systems. Mention that computer assisted displays (i.e. computer graphics) now provide a CAD checkpoint for the designer and serve as the hardcopy layouts for detailed part drawings or the CAM preparation of N/C machine input tapes. Mention that in the future, graphics will be used merely as visual display checkpoints, with the computer being used to bridge the gap from conceptual design to actual production without drawings. But emphasize that the importance of properly understanding graphical displays is increased, not diminished, in such situations.

3. Permit students to utilize computer graphics in other areas that reflect current viable industry/government applications. Among those are various automated drafting programs for two and three dimensional design and layouts. These are to be distinguished from the conceptual design approach in that rather than thinking in a model space, the designer/draftsman is steeped to commands for graphical imagery, i.e., the final drawings per se. Auxiliary to these might be the use of graph/chart/diagram packages.

4. If the situation permits it, utilize interactive computer graphics. Such a medium heightens the designer/draftsman's experience. For example, interactive graphics can allow one to see the step by step buildup of a model design, or it can permit the computer to "lead" the student through his design or graphics (i.e. "role playing").

5. With judicious care permit students to use computer graphics where clear cut educational benefits may be realized even if not associated with computer graphics per se. Such might be the development or reinforcement of more traditional concepts.

6. Avoid the use of computer graphics as mere "frosting on the cake" in demonstrations and assignments. Its use without these or additional goals will merely shortchange the student.

REFERENCES

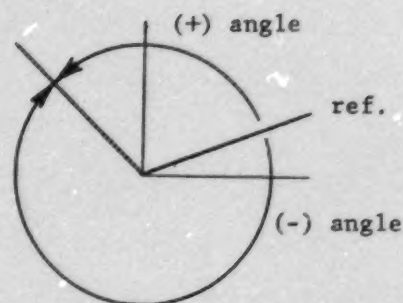
1. W. M. Newman and R. F. Sproull, Principles of Interactive Computer Graphics, McGraw-Hill, 1973.
2. H. R. Puckett, "Computer Method for Perspective Drawing," J. Spacecraft and Rockets, January, 1964, p. 44.
3. E. V. Mochel, "Computer Projections of Three-Dimensional Objects," Engineering Education, January, 1969, p. 413.
4. H. Peckham, "Computer Graphics in Physics," Proceedings of the 1972 Conference on Computers in Undergraduate Curricula.
5. H. H. Plott et al., "A Three Dimensional Display System," IEEE Trans. Education, Vol. E-17, No. 3, p. 152 (1974).
6. R. P. Hoelscher et al., Graphics for Engineers, Wiley and Sons, 1968. See Chapter 33.
7. W. J. Luzzader, Innovative Design with an Introduction to Design Graphics, Prentice-Hall, 1975. See Part 4.
8. "Computer Graphics for Designers," The University of Michigan Summer Conferences, Ann Arbor, 1967-- (Bound class notes.)
9. Anon., "CalComp Graphics Functional Software - USAS Fortran/Drafting," California Computer Products, Inc., Anaheim, 1969.
10. D. J. Logan, "Draft-Aid," Graphic Science, February, 1970, p. 11.

11. Anon., "Precision Artwork Language," Graphic Science, November, 1970, p. 12.
12. Anon., "Three-D, CalComp Applications Software," California Computer Products, Inc., Anaheim, 1970.
13. C. E. Hall, "Computer Graphics Programs," Department of Engineering Graphics, LSU (undated, circa 1971).
14. E. V. Mochel, "Introduction to 3D Programs," Summer School on Computer Graphics, sponsored by the Engineering Design Graphics Division of ASEE, 1972.
15. Anon., "TRIDM Manual" and related documents, Freshman Engineering Division, NCSU (undated, circa 1972).
16. B. Herzog and F. Shadko, "DRAWL 70: A Computer Graphics Language," The University of Michigan, Technical Rpt. 30, 1970.
17. M. H. Pleck, GRAFIX: "A User's Guide," Department of General Engineering, University of Illinois, Urbana-Champaign, 1970 (rev. 1974).
18. B. R. Fowler, "XYMASK," Bell Telephone Laboratories Record, July, 1969.
19. M. H. Pleck, "GATRAN: A User's Guide," Department of General Engineering, University of Illinois, Urbana-Champaign, 1968 (rev. 1970).
20. I. E. Sutherland, "Computer Displays," Scientific American, June, 1970, p. 57.
21. M. D. Prince, Interactive Graphics for Computer-Aided Design, Addison-Wesley, 1971.
22. N. O. Olesten, Numerical Control, Wiley-Interscience, 1970.
23. IIT Research Institute, APT Part Programming, McGraw-Hill, 1967.
24. N. H. Cook, "Computer-Managed Parts Manufacture," Scientific American, February, 1975, p. 23.
25. P. Villers, "Advanced Interactive Graphic Terminals for Computer Aided Design," Graphic Science, Oct. 1970, p. 15
26. L. H. Hazlett, "Design Center for Custom LSI," Graphic Science, February, 1971, p. 16.
27. D. L. Lossing and A. L. Eshleman, "Planning A Common Data Base for Engineering and Manufacturing," Paper 6300, Douglas Aircraft Company, Long Beach, 1974.

5.0 GRAFIX Drafting Routines

For each of the following routines, a description is presented with a simulated drafting statement (see 4.0) in which

1. The first element is the command code.
2. The remaining elements are arguments that represent values, where, as appropriate:
 - a. All distances are in plotter inches (up to but no more than 3 decimal places should be used).
 - b. All angles are in degrees and are considered positive in the counter-clockwise (ccw) sense. Negative angles are permitted.



The basic GRAFIX routine is presented first. It is concerned with the motion of the simulated drafting pen across the paper and the optional relocation of the (0,0) origin. It is written in the form of the statement:

5.1 P X,Y,N

P moves the drafting pen in a straight line from its previous location to the point (X,Y), where X and Y are coordinates on the plotter paper measured from the last (0,0) in inches. N is an integer which controls the pen as follows:

- N = 2: the pen contacts the paper while moving to (X,Y); a line is drawn.
- N = 3: the pen does not contact the paper while moving to (X,Y); no line is drawn.
- N = -3: same as N = 3 except that (X,Y) becomes the new (0,0).
- N = -2: same as N = 2 except that (X,Y) becomes the new (0,0).

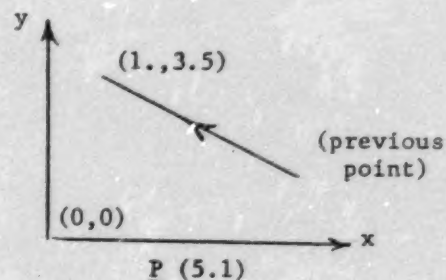
Examples:

1. P 1.,3.5,2

The command causes the plotter to draw a straight line from some previous position to the point (1.,3.5).

2. P 1.,3.5,-3

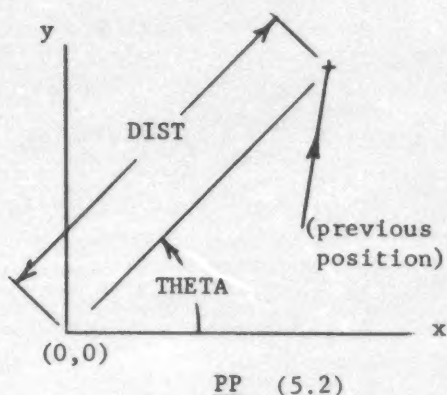
The pen moves as in example 1 except that no line is drawn and (1.,3.5) becomes the new (0,0).



The basic drafting pen movement described above is the basis of all other drawing routines. P can be repeatedly applied by more complex routines to form geometrical shapes including curves if the (X,Y) increments are small and closely spaced. Some of the commands described below employ this principle.

5.2 PP DIST,THETA,N

PP is like command P, but enables the pen to be directed in polar fashion. In its use, the end of the line is determined by its polar values DIST and THETA rather than by (X,Y). THETA is in degrees is measured from the x-axis. It is + or - according to rule 2b in Section 5.0. N has the same values and effects as it does in command P.



NSTART	NEND	Corresponding line
0	0	-----
1	0	-----
0	1	-----
1	1	-----

start end

HL,HLP,HL2 (5.3-5.5) →

5.3 HL NSTART,X,Y,NEND

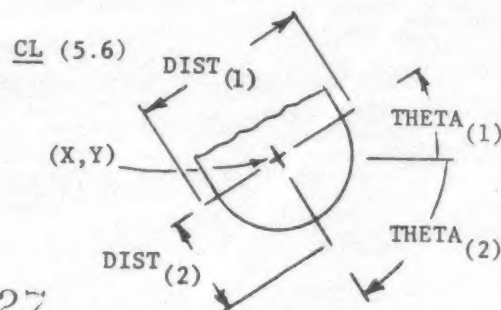
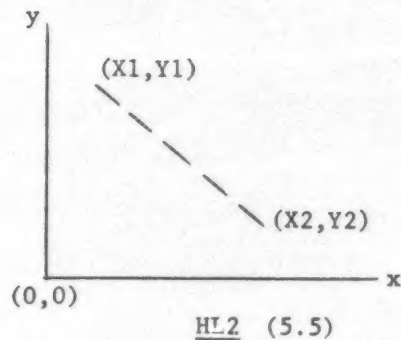
HL draws a hidden line from (0,0) to (X,Y). (X,Y) becomes a new (0,0) and the pen stops there. The table shown summarizes the end conditions, i.e., dash or space. NSTART and NEND are integers.

5.4 HLP NSTART,DIST,THETA,NEND

HLP is to HL what PP is to P. It does what HL does, but permits the end of the line to be specified by the polar values DIST and THETA rather than by (X,Y). THETA is as in PP. The end of the line becomes a new (0,0) and the pen stops there. NSTART and NEND are as in HL.

5.5 HL2 NSTART,X1,Y1,X2,Y2,NEND

HL2 is a supplement to HL and HLP. It is different in that both end points are specified, i.e., (X1,Y1) and (X2,Y2). The origin is not reset but the pen stops at (X2,Y2) when finished. NSTART and NEND are as in HL.

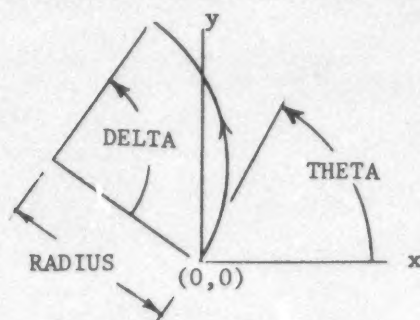


5.6 CL X,Y,DIST,THETA,N

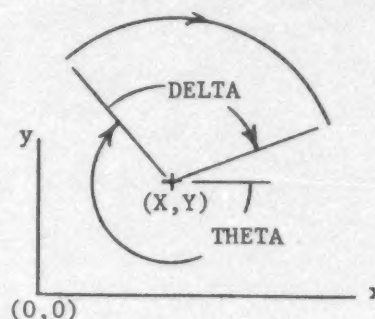
CL draws a centerline with a middle dash centered at (X,Y). The length of the centerline is DIST and its inclination is THETA measured from the x-axis. N, an integer, is normally 1. If N is 2, only the right half of the centerline (prior to rotation) is drawn; DIST in this case is the length of the right hand portion only. The previous (0,0) is not changed; the pen stops at (X,Y) when finished.

5.7 ARC1 THETA,DELTA,RADIUS,K

ARC1 draws a visible arc ($K = 1$) or hidden arc ($K = -1$) with a given RADIUS. THETA measured from the + x-axis, is the direction of a tangent to the arc at (0,0) pointing in the general direction the arc is to be drawn. DELTA is the angle the arc is drawn through. If negative, the arc will proceed in a cw direction; if positive, in a ccw direction. Thus DELTA represents the change of the inclination of the initial tangent. The end point of the arc becomes a new (0,0) and the pen stops there when finished.



ARC1 (5.7)



ARC2 (5.8)

5.8 ARC2 THETA,DELTA,X,Y,RADIUS,K

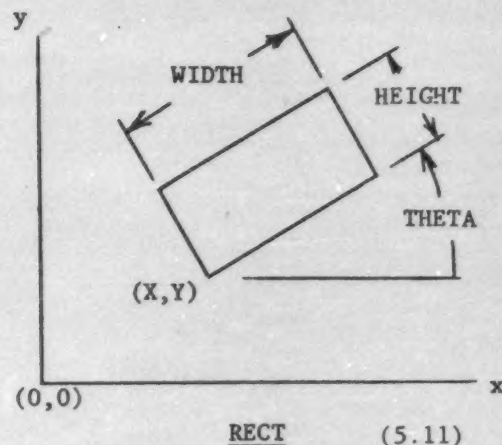
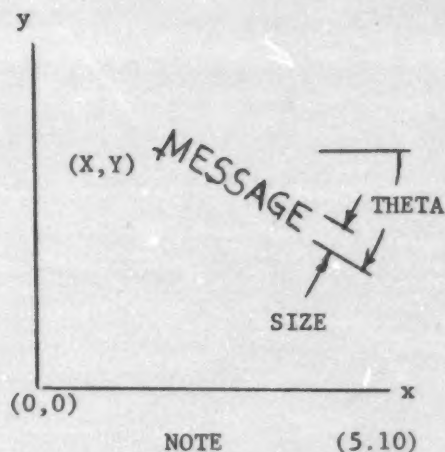
ARC2 draws the same type of arc as ARC1, but defines it differently. (X,Y) is the center of the arc and THETA is the inclination of the fictitious radial line intersecting the start of the arc. DELTA, RADIUS and K are as in ARC1. The previous (0,0) is not changed; the pen stops at the end of the arc when finished.

5.9 CIR X,Y,DIAM,K

CIR draws a hidden or visible circle with a diameter DIAM. As it is a special application of ARC2, X, Y and K are as defined in ARC2. (The pen stops on the circle at THETA = 0).

5.10 NOTE X,Y,SIZE,THETA,N,*MESSAGE

NOTE writes a message composed of any of the characters found on the keys of the card punch. (X,Y) are the coordinates of the lower left corner of the first character. SIZE is the desired height of the characters in the message. THETA is the angle at which the message is inclined from the x-axis. N is the number of characters, including blanks but not the initial *. It should not exceed 20 characters. If more than 20 are used, they will be lost. MESSAGE is the message, preceded by an * symbol. The previous (0,0) is not changed; the pen stops at the end of the message whose approximate length is $6 \cdot \text{SIZE} \cdot N / 7$.



5.11 RECT X,Y,THETA,WIDTH,HEIGHT

RECT draws a rectangle of base WIDTH and altitude HEIGHT and its lower left hand corner at (X,Y). The base of the rectangle may be inclined at the angle THETA, measured from the x-axis. The previous (0,0) is not changed; the pen stops at (X,Y) when finished.

5.12 ARH X,Y,THETA

ARH draws an 0.18" long arrowhead with its tip at (X,Y). THETA is the direction in which the arrow is pointing, measured from the x-axis with the usual sign convention. The previous (0,0) is not changed; the pen stops at (X,Y) when finished.

5.13 FCTR SCALE

FCTR scales the drawing of all commands that follow it by an amount SCALE. SCALE = 1. is full size, 0.5 is half size, etc. It may be used to scale a complete drawing if it is used before the pen is first moved. However, remember that it will affect the location of the whole drawing and change the size of messages written by NOTE.

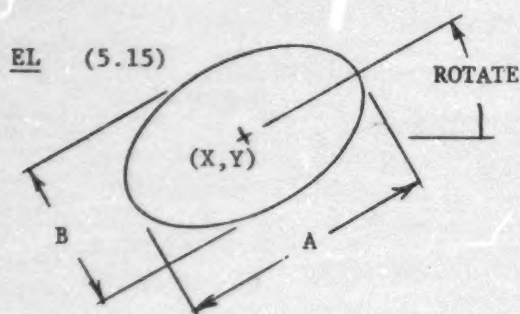
5.14 PSET IREFPT,N

PSET is a useful alternative to P and PP when it is desired to return quickly to up to twelve previous positions on the plotter paper. PSET must be used twice to achieve this result for any point:

1. First, the point IREFPT must be designated. When $N = 0$ and IREFPT is assigned the integer 0-11, this is achieved. The current physical position of the pen becomes known at point IREFPT.
2. Then, to use the previously designated point IREFPT, do either of the following:
 - a. Let $N = +2$ or $+3$ as in P and PP and a corresponding pen motion is made to IREFPT. Effectively, X, Y in P and DIST, THETA in PP are replaced by IREFPT.
 - b. Let $N = 1$ and IREFPT becomes a new (0,0) origin without any pen motion whatsoever. This is different from P and PP usage.

5.15 EL X,Y,A,B,ROTATE

EL draws an ellipse with its center at the point (X,Y). A and B are the major and minor axes respectively. ROTATE is the inclination of the major axis, measured in degrees, positive ccw from the +x-axis. Upon completion of the routine (X,Y) becomes a new (0,0) and the pen stops there. See the sketch below for use in isometric drawings.



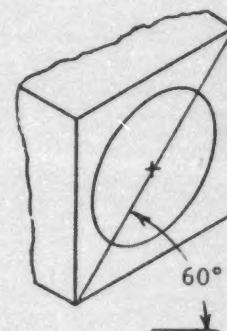
For circle in isometric drawing:

$$A = 1.224 * \text{diameter}$$

$$B = 0.707 * \text{diameter}$$

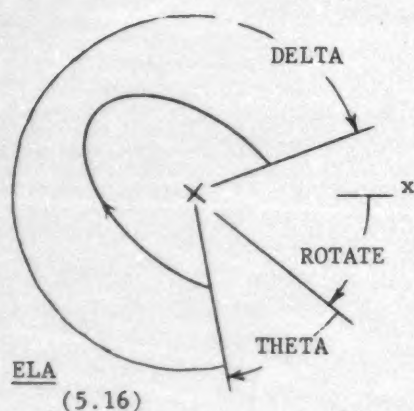
To get to (X,Y):

Use two PP steps, set (0,0) there, and use EL 0,0,...



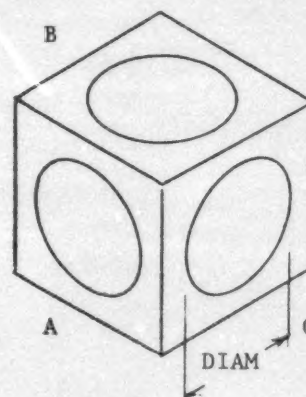
5.16 ELA X,Y,A,B,THETA,DELTA,ROTATE

ELA draws an elliptic arc that may be regarded as part of an ellipse drawn by routine EL. Hence X,Y,A,B and ROTATE are the same as in EL. Letting the right end of the major axis (prior to rotation) be a reference, THETA is the initial angle of the elliptic arc and DELTA is the change in angle. Upon completion of the routine, (X,Y) becomes a (0,0) and the pen stops there. (Figure 5.16)



NVHP	Case
-1	A
0	B
1	C

ISOC (5.17)

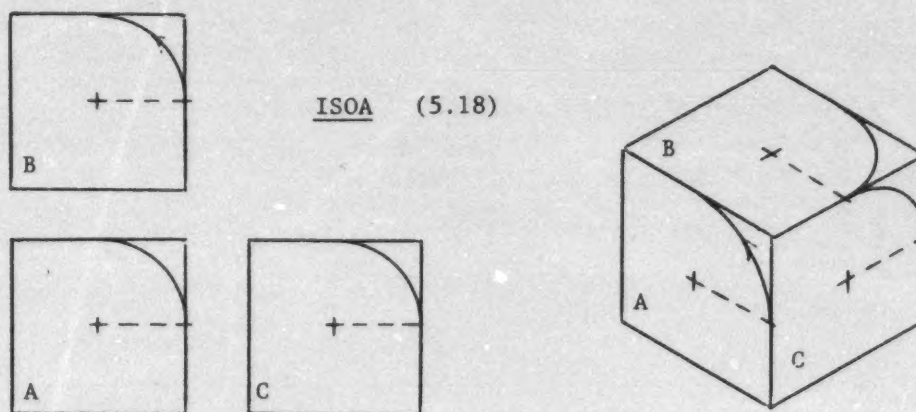


5.17 ISOC X,Y,DIAM,NVHP

ISOC draws orthographic view circles in the principal planes of an isometric drawing, i.e., ellipses. (X,Y) is the center of the circle which becomes an ellipse. DIAM is the diameter of the orthographic circle. NVHP can have three values -1,0,+1 corresponding to the vertical, horizontal and right profile (front, top, r. side) as indicated in the sketch. After completion (X,Y) becomes a new (0,0) and the pen stops there.

5.18 ISOA X,Y,RADIUS,THETA,DELTA,NVHP

ISOA draws arcs of orthographic view circles in the principal planes of an isometric drawing. They may be regarded as parts of isometric circles drawn by ISOC. Hence X,Y and NVHP are the same as in ISOC. RADIUS is the radius of the orthographic circle. THETA and DELTA define the isometric arc in terms of the angles associated with the orthographic arc. They are similar to THETA and DELTA in ARC2. The only trick in their use is to visualize the connection between the orthographic view and the isometric. This is perhaps best seen in the sketch. THETA is the initial angle of the arc, with respect to the dotted reference lines in the figure, but measured from the corresponding orthographic view. DELTA is the number of degrees the arc travels through. Both are positive in the ccw sense. In the figure, THETA = 0, DELTA = 90 are shown for NVHP = -1,0,1 respectively. Note that the apparent isometric arcs are not 90°. After completion (X,Y) becomes a new (0,0) and the pen stops there.



6.0 Comments in the Printed Output

It is often helpful in reading a printed output to have comments describing what is being done. A comment of any length may be placed in the program listing in two ways. First, they may be placed right after the last item of a routine command, providing they are preceded by a comma and an * as in the *MESSAGE of the ID card and NOTE. Second, they may be placed on separate cards. In doing so, the user may think of them as non-functional routines: Use the code *, leave at least one blank and begin the message without the usual *. Several excellent examples of comment usage are provided in the example problems.

APPENDIX 2

GATRAN Routine Descriptions

CALL START

This routine must be used at the start of every program. It initializes the storage of point and line designations. It can be used later in the program when it is desired to start afresh. It has the effect of erasing all previous information.

where NAMEPT is a name associated with the set of points of a specially defined object. It must obey the rule for naming integer variables (1-6 letters and numbers, the first being I, J, K, L, M, or N).

NAMELN is a similar name associated with the set of lines in the object.

NP is the number of points in the object. It is an integer constant.

NL is the number of lines in the object, also an integer constant.

This routine is used to define the points and lines of a special object by means of punched data cards. Imagine that the points are numbered 1 to NP. Then each point is described on a data card by putting the X coordinate in columns 1-10, the Y coordinate in columns 11-20, and the Z coordinate in columns 21-30. Decimal points must be used. Since the point numbers are not associated with the cards, they must be in the order 1 to NP. Accordingly, one may place identifying numbers anywhere after column 30 for purposes of keeping the cards in correct order (the computer will not read the identifiers). The points, taken in the above order, are then referred to by the NL line definitions. Each line is described on a data card by a beginning and end point, referring to the order given above. The first point is in columns 1-10 and the second in 11-20. No decimal points are used and the numbers must be right justified (i.e., ending in columns 10 or 20 respectively). The set of line cards is put behind the set of point cards and may have identifiers anywhere beyond column 20. Usage restriction: There may be no more than 200 points and 300 lines on data cards for all DEFINE uses combined.

CALL IMAGE (NAMEPT,NAMELN,NP,NL,XSCL,YSCL,ZSCL,RY,RZ,RX,XTRN,YTRN,ZTRN)

where NAMEPT, NAMELN, NP, NL are as described as in routine DEFINE.

XSCL, YSCL, ZSCL are multiplicative scale factors to be used to change the original size of an object in its component x,y,z directions.

RY, RZ, RX are the rotations of the object in degrees about the y,z,x axes from its defined position. The rotations must be performed in RY, RZ, RX order to obtain the correct position.

XTRN, YTRN, ZTRN are translations associated with the movement of the object in the x,y,z directions from its defined position.

This routine is used to place an object, which has been previously specified by a DEFINE call, somewhere in the coordinate system. To obtain the correct object, assume scaling and movement will be performed in the following sequence: scaling, rotation, and translation. This is the order given in the list of arguments, which are all floating point numbers with the exception of the first four. Usage restriction: there may be no more than 1000 points and 1500 lines constructed for all uses of IMAGE, BLOCK and CYLIN combined.

CALL BLOCK (WIDTH,HEIGHT,DEPTH,RY,RZ,RX,XTRN,YTRN,ZTRN)

where WIDTH, HEIGHT, DEPTH are the dimensions of a right rectangular prism corresponding to the x,y,z directions before it is moved from its predefined position.

RY, RZ, RX are its rotations about the y, z, x axes in degrees.

XTRN, YTRN, ZTRN are its translations in the x, y, z directions from its predefined position.

This routine is used to place a right rectangular prism, (eight points, twelve lines in total), somewhere in the coordinate system. Before movement the block is imagined to be predefined with one of its corners at the origin and its sides extending along the +x, +y, +z axes respectively as in Figure 3. In specifying the movement, assume rotation before translation to obtain the correct position. All the arguments are floating point values. Usage restriction: there may be no more than 1000 points and 1500 lines constructed for all uses of IMAGE, BLOCK and CYLIN combined.

CALL CYLIN (RADA,RADB,HEIGHT,RY,RZ,RX,XTRN,YTRN,ZTRN)

where RADA and RADB are the radii associated with the top and the base of a circular cylinder respectively in its predefined position.

HEIGHT is its vertical height.

RY,RZ,RX are its rotations about the y, z, x axes in degrees.

XTRN,YTRN,ZTRN are its translations in the x, y, z directions from its predefined position.

This routine is used to place a right circular cylinder or cone represented by twelve lateral elements (twenty-four points, thirty-six lines in total), somewhere in the coordinate system. Before movement the cylinder or cone is imagined to be predefined with its axis coinciding with the y coordinate axis and its base on the x-z plane as in Figure 4. In specifying the movement, assume rotation before translation to obtain the correct position. All the arguments are floating point values. Usage restriction: there may be no more than 1000 points and 1500 lines constructed for all uses of IMAGE, BLOCK and CYLIN combined.

CALL COUNT

This optional routine is used to write a message in the printed output that contains the totals of lines and points utilized at the time the call is made. Two sets of totals are printed: one for all DEFINE uses; the other for the combined uses of IMAGE, BLOCK and CYLIN. Remember that there are maxima of points and lines that may be used. COUNT may be called as often as desired.

CALL SINW (RY,RZ,RX,SF)

where RY, RZ, RX are rotations of the constructed object about the y, z, x axes in floating point degrees.

SF is the multiplicative floating point scale factor applied to the coordinates of the constructed object.

This routine draws a single orthographic view of the constructed object. The sight line is toward the origin along the z axis (Figure 1). Prior to drawing, the (0,0) plotter origin, corresponding to (0,0,0) of the object, must be reset by the user to a convenient position within the plotter boundaries, and when finished, reset about 8 inches to the right and repositioned vertically in preparation for the next view. In using the scale factor, note that one object coordinate unit will equal one inch on the plotter paper prior to scaling.

CALL ORTHO (RY,RZ,RX,SF)

where RY, RZ, RX are rotations of the constructed object about the y,z,x axes in floating point degrees.

SF is the multiplicative floating point scale factor applied to the coordinates of the constructed object. If it is positive, reference lines are drawn between orthographic views; if it is negative, reference lines are omitted (and absolute value used for scaling). If SF is set to 0, the user is in effect asking the routine to determine a scale factor that will best fit the views of the object within the viewing area. Reference lines are drawn in this case.

This routine draws three standard orthographic views of a constructed object. These are the top, front, and right side views, corresponding to sight lines looking into the origin along the y, z, and x axes. Reference lines are optional and the views are scaled and placed in projection. After the three view drawing is completed, the plotter origin is automatically moved ten inches to the right for the next drawing. Prior to this call, the user should insure that the plotter origin is located at the mid-height of the plotter paper ($y=5''$). Otherwise the multiview drawing will not be vertically centered and may cross the plotter boundary. This vertical centering is done automatically if a call to ORTHO or PERSP (but not SINW) precedes the current call. In using the scale factor, note that one object coordinate unit will equal one inch on the plotter paper prior to scaling.

CALL PERSP (RY,RZ,RX,SDR,XL,YL)

where RY, RZ, RX are the familiar rotations of the constructed object about the y, z, x axes in degrees.

SDR is the perspective sight-to-distance ratio. It controls the location of the sight point and the resulting perspectivity. A value of 1.5 to 2.0 will give a normal perspective. Larger values will tend to produce near orthographic projections and smaller values very pronounced perspectives. It is a floating point value.

XL, YL are the sizes of the drawing desired in the x and y directions in inches. They result in automatic scaling of the perspective which is centered on the plotter paper. They are floating point values.

This routine draws a perspective view of the constructed object. The sight point is on the z axis as in Figure 2 and the object is imagined to be automatically recentered about the z axis with its most forward point (z direction) imagined to touch the picture plane. Upon completion the plotter origin is moved XL+1 inches to the right for the next view. Prior to this call, the user should insure that the plotter origin is located at the mid-height of the plotter paper ($y=5''$). Otherwise, the perspective view will not be vertically centered and may cross the plotter boundary. This vertical centering is done automatically if a call to ORTHO or PERSP (but not SINW) precedes the current call.

CALL LABEL (RY,RZ,RX,SF,XL,YL)

where RY, RZ, RX are the rotations used in the view routine LABEL is applied to.

SF is the multiplicative scale factor used in the view routine LABEL is applied to.

XL, YL are the x and y direction sizes of the border lines to be drawn around the view.

This routine draws a border around a view, centered about the last plotter origin. In the upper right corner the supplied rotation angles and scale factor are printed. LABEL is automatically called by ORTHO; it may be called optionally with SINW or PERSP. (With PERSP, use it just preceding the call.)